Toxic Comment Classification Challenge

by Team 557_C2 (Jiachen Liu and Haomiao Ni)
Problem Description

Background:

Toxic comments (rude or disrespectful comments) can make people stop expressing themselves and give up on seeking different opinions.

For effectively facilitate online conversations, some communities began to limit or even completely shut down user comments.
Problem Description

Objective:

Building models to detect and classify toxic comments

Our task:

Classify multi-label toxicity comments (6 classes) from a large-scaled dataset (more than 150k sentences)
Dataset

Training set: 159,571 comments, each with 6 class labels

Testing set: 153,164 comments without labels

Evaluation Metrics

Submissions are now evaluated on the mean column-wise ROC AUC. In other words, the score is the average of the individual AUCs of each predicted column.
Data Pre-processing

1. Change all the comments to low-cased
2. Remove all the punctuation and digits
3. Split each word
4. Clip or pad each comment to make sure that their length is fixed (400)
Word Vector Representation

We directly use pre-trained word vectors to represent each word

crawl-300d-2M.vec: 2 million 300-dim word vectors trained on Common Crawl

The final size of our dictionary: 35,933
Model Architecture

- Linear layer and final classification layer
- Max pooling to get global feature
- 1D CNN to get word-context information
- Bidirectional GRU
- Dropout=0.3

Input Batch vectors

<table>
<thead>
<tr>
<th>Layer Description</th>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 * 6-dim</td>
<td><img src="image1.png" alt="Diagram" /></td>
</tr>
<tr>
<td>1 * 128-dim</td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
<tr>
<td>ReLU+Maxpooling</td>
<td><img src="image3.png" alt="Diagram" /></td>
</tr>
<tr>
<td>399 * 128-dim</td>
<td><img src="image4.png" alt="Diagram" /></td>
</tr>
<tr>
<td>CNN</td>
<td><img src="image5.png" alt="Diagram" /></td>
</tr>
<tr>
<td>400 * 256-dim</td>
<td><img src="image6.png" alt="Diagram" /></td>
</tr>
<tr>
<td>h_1, h_2, ..., h_N</td>
<td><img src="image7.png" alt="Diagram" /></td>
</tr>
<tr>
<td>GRU</td>
<td><img src="image8.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Dropout</td>
<td><img src="image9.png" alt="Diagram" /></td>
</tr>
<tr>
<td>400 * 300-dim</td>
<td><img src="image10.png" alt="Diagram" /></td>
</tr>
<tr>
<td>w_1, w_2, ..., w_N</td>
<td><img src="image11.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>
Parameter Setting

Cross-validation:

10-fold cross-validation to train & evaluate the model respectively

training settings:

Adam optimizer with reduce_lr_onplateau scheduler

Original leaning rate 0.001 and batch-size 512

Model selection and ensembling:

During training epochs we save the checkpoint if the validation auc is better than the previous models

After training we use the best model to predict the testing results for each fold

We ensemble(take average) all the 10-fold testing results as our final prediction
Model Selection and comparison with ML methods

Our baseline model (without ensembling):

Model without dropout:

Model with average pooling:

Model with single-direction GRU:

Model with random forest:
## Parameter tuning and results comparison

Our baseline (without ensembling):

```
test_0.csv  
2 days ago by jcliu0428
add submission details
```

### Training with larger batch size (1024):

```
test_with_bs_1024.csv  
a minute ago by jcliu0428
add submission details
```

### Training with smaller batch size (256):

```
test_with_bs_256.csv  
8 minutes ago by jcliu0428
add submission details
```

### Training with smaller learning rate (1e-4):

```
test_with_lr_1e-4.csv  
just now by jcliu0428
add submission details
```

### Training with larger learning rate (1e-2):

```
test_with_lr_1e-2.csv  
a few seconds ago by jcliu0428
add submission details
```
Other insights and summary

Baseline:

1. Learning rate is more significant than other hyper-parameters

2. Data cleaning and feature selection is much more important than model selection

Training using data without cleaning:

3. Ensembling is all you need
Reference